Comparison of Object Detection Algorithms for Livestock Monitoring of Sheep in UAV images

Oliver Doll[1,*] *Alexander Loos*[1]

[1] *Audio-visual Systems, Fraunhofer IDMT, Ehrenbergstr. 31, 98693 Ilmenau, Germany*
** E-mail: oliver.doll@idmt.fraunhofer.de*

Abstract: This paper presents the EU funded project *SPADE*, a European initiative that aims to create an Intelligent Ecosystem utilizing unmanned aerial vehicles (UAVs) for delivering sustainable digital services to various end users in sectors like agriculture, forestry, and livestock. The project's main goal is to cater to multiple purposes and benefit a wide range of stakeholders. In this paper we specifically concentrate on the livestock use-case and explore how state-of-the-art computer vision algorithms for object detection, tracking, and landscape classification, deployed on edge devices in drones, can offer researchers, conservationists, and farmers a non-intrusive, cost-effective, and efficient method for monitoring livestock increasing animal welfare, and optimize livestock management. We present initial findings by comparing the performance of different state-of-the-art object detectors on publicly available UAV images of sheep. The key performance metrics used are average precision, mean average precision and mean average recall. These findings should enable a better pre-selection of potential object detectors for the presented edge device use case.

1 Introduction

Unmanned aerial vehicles (UAVs) equipped with automated animal detection systems and integrated with advanced computer vision algorithms have emerged as a promising solution for wildlife monitoring, conservation, and livestock farming. This technology provides researchers, conservationists, and farmers with a non-intrusive, cost-effective, and efficient method for monitoring livestock, ultimately enhancing animal welfare and optimizing livestock management.

The strategic objective of the European initiative *SPADE*[*] is to develop an Intelligent Ecosystem that leverages UAVs to deliver sustainable digital services across multiple sectors, including agriculture, forestry, and livestock. This paper provides a comprehensive overview of the project's objectives. While all three use-cases (agriculture, forestry, and livestock) are covered, the livestock pilot receives particular emphasis. Therefore, the following sections focus on providing a state-of-the-art review of object detection algorithms in UAV footage, their implementation on edge devices, and how modern deep learning technology can assist farmers in optimizing livestock management. In order to get a feeling of how well state-of-the-art object detectors are suited for the special task of detecting and counting herds of sheep from UAV images, we trained and evaluated a number of object detectors for the task at hand. We thoroughly compare their performance using the metrics Average Precision (AP), mean Average Precision (mAP) and mean Average Recall (mAR) while considering image input size and complexity of the detectors. Based on these results we are identifying potential strengths and shortcomings, and assess their suitability to be implemented on an edge device. This allows for more extensive experiments on a much smaller set of object detectors in future work.

The paper is organized as follows: In section 2 we first give a brief overview of state-of-the-art algorithms for object detection. In particular, we first focus both on general object detection and then object detection from aerial images. We then review approaches for the detection of animals from drone footage as well as approaches for the classification landscapes from UAV images. Although this paper's main focus is on the *SPADE* livestock-use case, we briefly introduce related work on the other two pilots, agriculture and forestry, as well. In section 3 we introduce the methodology of the preliminary experiments we conducted by training and evaluating a set of state-of-the-art object detectors on a publicly available image dataset of sheep recorded from UAVs, followed by the obtained results in section 4. We conclude the paper in section 5 giving a short summary as well as an outlook on what we plan as future work. To enable reproduction of results we made configuration files and instruction for training available at `https://github.com/idmt-odoll/sheep_detection_in_UAV_images`.

[*] `https://spade-horizon.eu/`

2 Related Work

2.1 General Object Detection

Automatic object detection has been extensively studied in the field of computer vision, particularly with the advent of deep learning and large annotated datasets. The detection and classification of general objects have gained significant popularity in image and video processing. One notable early attempt to utilize convolutional neural networks (CNNs) for object detection was the Regions with CNN features (R-CNN) algorithm, introduced by Girshick et al. in 2014 [13]. This algorithm presented a straightforward and scalable approach that improved mAP by over 30 % compared to previous methods in the PASCAL Visual Object Classes Challenge, a widely recognized object recognition competition.

Since then, there have been several enhancements and refinements to the basic R-CNN algorithm. Fast R-CNN [12] and Faster R-CNN [27] were developed, building upon the initial approach. In 2015, a real-time capable object detection framework named You Only Look Once (YOLO) was introduced by Redmon et al. [26]. YOLO, known for its exceptional speed, introduced more localization errors but exhibited a lower likelihood of predicting false positives in the background. Over the years, further improvements and iterations of YOLO, such as YOLOv8 [17], have been developed, constituting a family of object detection architectures and models.

Around the same time, another one-stage detector called Single-Shot-Detector (SSD) was proposed in [21]. SSD achieves real-time performance by utilizing feature maps of various scales and aspect ratios, enabling the capture of objects at multiple resolutions within a single neural network.

More recently, EfficientDet [30] was introduced as a family of object detectors aiming to strike a balance between accuracy and efficiency. EfficientDet employs EfficientNet [29] as its backbone, a high-performance CNN with fewer parameters, resulting in reduced computational resource requirements. Through a compound scaling method that optimizes both the width and depth of the network, EfficientNet achieves comparable accuracies on traditional classification tasks compared to competing network architectures like ResNets.

Another family of anchorless one-stage object detectors is CenterNet [10]. Unlike traditional approaches that focus on predicting bounding box coordinates or keypoints, CenterNet simplifies detection by concentrating on identifying object centers rather than precise bounding box coordinates. Similar to other object detectors, CenterNet employs a second prediction head to estimate regression values for box dimensions and offsets from the predicted center point. One advantage of CenterNet is its replacement of the post-processing step of Non-Maximum Suppression (NMS) with a more efficient algorithm that can be integrated directly into the CNN. This integration enables much faster inference compared to competing methods.

Overall, these advancements in object detection algorithms have significantly contributed to the progress of the field, with each approach offering unique characteristics and trade-offs in terms of accuracy, speed, and simplification of the detection process.

2.2 Object Detection in Aerial Images

Recent years have witnessed significant advancements in automatic object detection and tracking in UAVs, owing to the remarkable capabilities of state-of-the-art object detectors and the emergence of edge computing devices. Object detection models such as Faster R-CNN, YOLO, and SSD have become the foundation for detecting objects in aerial images. However, detecting small, oriented objects with high accuracy in UAV imagery, given the high altitudes at which UAVs typically operate, necessitates various adjustments and innovations.

A comprehensive overview of cutting-edge object detectors specifically designed for small objects and object localization in UAV images was presented in [4] and [35].

In [39], an intriguing approach was introduced to address the challenges posed by objects captured at different scales due to varying altitudes and motion blur resulting from the high-speed and low-altitude flight of drones. The authors proposed TPH-YOLOv5, an enhanced version of YOLOv5, which introduced an additional prediction head for detecting objects at different scales. Moreover, the original prediction heads were replaced with Transformer Prediction Heads (TPH) that leverage a self-attention mechanism to enhance object detection capabilities. To identify attention regions in scenarios with dense objects, the authors integrated the convolutional block attention model (CBAM) into the model architecture.

The authors demonstrated that TPH-YOLOv5 exhibits exceptional performance when applied to drone-captured scenarios, outperforming competing methods. These findings highlight the effectiveness of the proposed enhancements in addressing the specific challenges associated with object detection in UAV imagery.

2.3 Animal Detection in Aerial Images

Although, general object detection is getting not only more accurate but also more efficient, detecting animals in UAV images is still a challenging problem. Especially in precision livestock farming or monitoring of endangered species, high accuracy is demanded while smaller and more energy efficient models are needed due to implementation on mini-computers and edge devices. Recently, a lot of work has been done on the detection of animals in UAV imagery.
The authors of [7] compare YOLOv4 and YOLOv5 models to counted bovine cattle in images taken at altitudes of 20, 40, 80 and 100 m. All variants of YOLOv5 exceeded a precision of 92 % with the smallest model reaching a precision of 96 % and the largest model 98 %. An interesting finding was that the precision not necessary increases with the complexity of the model, in this case the YOLOv5-m model performs worse than the YOLOv5-s model. In [33], Wang et al. used the newer YOLOX nano model and improved the detection performance for small objects by enhancing the CSP-Darknet backbone and introducing a weighted aggregation feature re-extraction pyramid module as neck of the model. The obtained mAP for cattle, sheep and horses was 86.47 % at a altitude of 300m. They also did experiments on the scale adaptability of the model to object scales that differ from the training data. For increasing scale differences the performance decreases but different animal types are impacted differently by shrinking and expanding.
In case of common cranes, it was shown in [3] that the use of automatic approaches to count individuals in UAV images can be more accurate than manual counting of field observers, who underestimated the population. The increased accuracy opens up more accurate monitoring of animal herds and populations. The applied YOLOv3 model reached a precision of 99.91 % and a recall of 94.59 % for RGB images at daylight. In [25] two YOLOv4 models, YOLOv3 and SSD with MobileNet backbone were utilised to detect deer that are well adapted to their environment. From the tested models YOLOv4 achieved the best result with an precision of 86 % and recall of 75 %.
The Authors of [6] took a completely different approach by using a segmentation algorithm. They determined the species-specific sRGB-colour profile of adult Arabian Oryx and used that to segment patches of this certain colour profile. This approach achieved a precision of 100 % and recall of 98.87 %.

2.4 Landscape and Grassing Region Analysis

Automatic landscape classification, alongside animal detection and tracking, can bring several advantages to livestock farmers. By analyzing the vegetation and land characteristics, farmers can optimize grazing management, strategically allocate grazing areas, and maximize the availability of high-quality forage for their livestock. This technology empowers farmers to make informed decisions that enhance productivity, animal welfare, and environmental sustainability on their farms.

While most publications in landscape classification focus on multi-spectral high-resolution satellite images (HRSI), there have been recent attempts to develop deep learning-based approaches specifically for UAV images. Wang, for example, presents a modified U-Net architecture that incorporates asymmetric convolutional blocks, a state-of-the-art attention mechanism, and a fully connected conditional random field [32]. This approach achieved promising results on a self-created hyperspectral image dataset, with an F1-score of 0.836. However, the complexity of the proposed model architecture renders its implementation on edge devices impractical.

Another interesting approach, as presented by Fan and Lu in [11], utilized a simplified AlexNet CNN architecture for landscape classification. They introduced a spatial and spectral feature fusion paradigm, which improved crop classification accuracy, raising their dataset's accuracy from 86.07 % to 92.76 %.

For a comprehensive overview of methods for landscape classification from UAV images, [1] provides valuable insights and information.

2.5 Computer Vision for Smart Forest Monitoring

Computer vision algorithms for object detection and segmentation in combination with UAVs are nowadays also used for monitoring and management of forests. Such technologies can enable real-time monitoring of forest ecosystems, including tree health, growth patterns, and environmental conditions. This valuable information can help forest managers to make informed decisions regarding forest management practices, such as identifying areas prone to disease or pests, optimizing harvesting strategies, and assessing the impact of climate change.

For instance, Puliti et al. [24] used drone laser scanning data (UAV-LS) in combination with a YOLOv5 object detector to measure the vertical positions of branch whorls and used this information as a proxy to derive height-growth information of individual trees.

Getting reliable information on tree height-growth dynamics is essential for optimizing forest management and wood procurement. Although, due to the small number of annotated training images and instances, the obtained results indicated a relatively poor performance of the YOLOv5-based whorl detector on single images, with the adoption of a multi-view approach and consequent post-processing of the detected whorls, the authors were able to increase the precision and recall score by an significant amount. The same authors later proposed to use a YOLOv5 object-detection model applied to UAV images to detect forest snow damage in [23], which presently relies on labor-intensive field surveys that potentially may introduce biases. Thus, automating this process by means of drones and modern computer vision algorithms is of high interest for forest owners and rangers.

For further information about tree classification and segmentation using computer vision and UAV data the interested reader is referred to Chehreh et al. [2] who give a thorough overview of state-of-the-art techniques in the field of smart forest monitoring.

2.6 Computer Vision for Precision Agriculture

Computer vision algorithms on drones enable farmers to monitor and manage crops more effectively, optimizing resource usage, such as water and fertilizers, and identifying areas requiring attention, such as pest infestations or nutrient deficiencies. One use case of computer vision in precision agriculture is the automatic counting of fruit flies, as infestation of fruit flies can endanger the harvest significantly. One way to monitor the fruit fly population is to set respective traps and count the fruit flies trapped. Depending on the environment and size of the farm, traps can be hard to reach and the manual counting is very time consuming and labour intensive. Being able to count the fruit flies from image and video data would decrease the effort. One approach to automate this process was presented in [28] by developing an electronic trap that also transmits real-time images of the trap surface to a server so that the fruit flies can be counted remotely. The authors of [15] have gone one step further by using a machine learning model to count the fruit flies from these images automatically. They identify the differences between two subsequent images and feed the regions of interest into their multi-attention CNN network with ResNet50 backbone. Another approach is presented in [8] by using a Faster RCNN model with ResNet50 backbone to count the fruit flies on these trap images. The model was trained on images from laboratory colonies and reached precision results of 93 % to 95 % on images in field conditions.

2.7 Edge Computing

The rise of edge computing and the integration of powerful processing capabilities into modern UAV systems have led to a growing focus on conducting object detection and analysis directly onboard the UAVs. Edge computing offers several advantages for deep learning applications in this context. Firstly, edge computing significantly reduces latency and response time by performing computations directly on edge devices or embedded systems. This enables real-time decision-making, making it particularly well-suited for time-sensitive applications such as real-time surveillance of livestock. Secondly, edge computing reduces dependence on cloud connectivity, ensuring that deep learning models can operate even in remote or disconnected environments. This is particularly valuable in scenarios where network connectivity may be limited or unreliable. Thirdly, edge computing minimizes bandwidth and storage requirements by processing data locally. Deep learning models can be deployed directly on edge devices, eliminating the need to transmit large amounts of raw data to the cloud for processing. This not only leads to significant cost savings but also optimizes resource utilization, as only relevant information needs to be transmitted.

In recent years, the industry has developed several edge devices specifically tailored for deep learning applications. Notably, the NVIDIA Jetson family, including models like Jetson Nano, Jetson Xavier, and Jetson TX2, has gained popularity as a powerful edge computing platform designed to accelerate deep learning applications at the edge. Another notable platform is Google's Coral, which consists of hardware components equipped with Google's Edge TPU and software tools for deep learning inference on the edge.

3 Methodology and Experimental Design

3.1 Description of Dataset

Large, annotated datasets are a prerequisite to properly train and evaluate object detectors for animal detection in UAV images. To compare different approaches and their ability to detect sheep in UAV images, we currently investigate a dataset called SheepCounter [22], which consists of 1727 images containing nearly only white sheep mainly on meadows. The images have a resolution of 3840 x 2160 pixel and and were all taken in bright daylight. A more detailed summary of the dataset is presented in Table 1. The split into train, valid and test set is 70 %, 20 % and 10 %. Further the object sizes are categorized based on the Common Objects in Context (COCO) metrics into small, medium and large objects. The share of small objects is vanishingly small at 0.4 % while large objects make up 82.38 % of the total amount. The share of large objects can be attributed to the high image resolution.

Table 1 SheepCounter dataset

	train	valid	test	all
number of images	1203	350	174	1727
instances per image	32.0	32.39	32.2	32.1
number of instances	38495	11337	5603	55435
small	147	39	32	218
medium	6209	2359	984	9552
large	32139	8939	4587	45665

3.2 Object Detectors

In this paper we train and evaluate a number of publicly available state-of-the-art object detectors on the *SheepCounter* dataset. We started by re-training object detector from the *YOLO*-family, starting from *YOLOv5* [16] up to the most recent implementation *YOLOv8* [17]. *YOLOv5* differs from older versions by using a PyTorch implementation instead of Darknet and introducing mosaic augmentation and auto learning bounding box anchors. *YOLOv6* [18, 19] builds upon *YOLOv5* and decouples detection and classification head. Furthermore the CSPNet backbone is replaced by EfficientRep and the PANNet neck is replaced by Rep-PAN. *YOLOv7* [31] applies extended efficient layer aggregation networks and other training tweaks to improve the training. *YOLOv8* is based on *YOLOv5* again and introduces more efficient convolutions and a decoupled head and other tweaks. With the development of *YOLOv8* an enhanced version of *YOLOv5* called *YOLOv5u* is released that uses the new anchor-free head of *YOLOv8*. All versions of YOLO we used in our experiments were pre-trained on COCO [20].

On the other hand, we compare competing approaches such as *SSD* [21], *EfficientDet* [30], *CenterNet* [10], and *FasterRCNN* [27] by exploiting the TensorFlow Object Detection API [14]. For each model we used the pre-trained COCO weights which are available in the TensorFlow 2 Detection Model Zoo.

3.3 Transfer Learning

All detectors mentioned in 3.2 were transfer-learned on the training set of the *SheepCounter* dataset to re-train them for the task at hand. For each detector we used the configuration file provided by the respective official repository. We left the main parameters as recommended in the according configuration file and trained until convergence by monitoring the loss on the validation set during

training. Since the standard configurations of different detectors can vary greatly, there is no common training routine. For example, *YOLOv8* detectors use translation, scaling, horizontal flipping and mosaic augmentation and also augment hue, saturation and value of the HSV colour space, while *EfficientDet* only uses horizontal flipping and random scaling and cropping as augmentation. Although it can be assumed that these differences have an influence on the results, the investigation of the augmentations on the performance of the detectors is not the scope of this work. After training, we freezed the model and inferred on the test set and evaluated the results by using the metrics explained in section 3.4.

3.4 Performance Metrics

For evaluation we utilized the PyCOCOtools package [20] which is a Python library that provides a set of utility functions and classes for working with the COCO dataset format. The library includes functionalities for performing evaluation of object detection and segmentation results against ground truth annotations, using standard COCO metrics such as AP, mAP, and mAR. For all COCO metrics at most 100 detections per image are considered and the mean value over multiple intersection over union (IoU) thresholds in $IoU = 0.5, 0.55, ..., 0.95$ is calculated. The average precision is calculated for $IoU = 0.5$. Besides the performance metrics, the detector input size and complexity are indicated for a better categorization of the results. Complexity is specified as necessary floating point operations per second (FLOPS) for one forward pass.

4 Results

The results of all trained detectors are depicted in Table 2. The detectors are ordered by detector family first and then by complexity. A first noticeable observation is that for six out of nine methods the biggest model does not achieve the highest mAP. The main reason for this is presumably the rather small size of the used dataset, so those large models are probably suffering from underfitting. For a more balanced comparison, a larger dataset must be taken into account. Despite the underfitting the *EfficientDet*, *CenterNet* and *Faster RCNN* models generally performed worse then the competing methods, as their best mAP are 0.491, 0.488 and 0.505 respectively. Leaving these three detector families, *YOLOv5* and *YOLOv6* out, the mAP depending on the complexity is plotted in Figure 1. *YOLOv5* and *YOLOv6* were not included in the plot as they could not show any advantage over the other detectors.

The highest mAP overall is reached by *SSD ResNet50 v1 FPN* with $mAP = 0.606$. With a complexity of 402.8 billion FLOPS, this model is also one of more complex models tested. It is about 4 times as complex as the second best model, *YOLOv7*, which achieves a $mAP = 0.574$ at 104.7 billion FLOPS. For very small models with complexity of 20 billion FLOPS or less, *YOLOv5-nu* (0.544) and *YOLOv8-n* (0.546) set themselves apart with a clear lead of about 0.04 in mAP over the next best model , *SSD MobileNet v2 FPN-lite*.

Looking at the average precision at $IoU = 0.5$, the gap between the best and second best detectors is smaller. Just like with mAP, *SSD ResNet50 v1 FPN* (0.959) has the highest average precision beating *YOLOv7* (0.955) and *YOLOv5-lu* (0.955) by a minimal difference of 0.004. For the very small models *YOLOv5-nu* (0.936) and *YOLOv8-n* (0.936) again stand out from the rest beating the next best detector by 0.01 (*SSD MN v2 FPN-lite*).

Regarding the mAR, the best detector again is *SSD ResNet50 v1 FPN* with a $mAR = 0.676$. The four *SSD* models with *ResNet* backbone reach the highest mean average recall beating all other detectors, followed by *YOLOv5-lu* (0.639) and *YOLOv7* (0.637). For very small models with a complexity of 20 billion FLOPS or less, *YOLOv5-nu* (0.61) and *YOLOv8-n* (0.612) again are the top two models. As for the mAP, the detectors based on *EfficientDet*, *CenterNet* and *Faster RCNN* also do not reach the performance level of the remaining detectors.

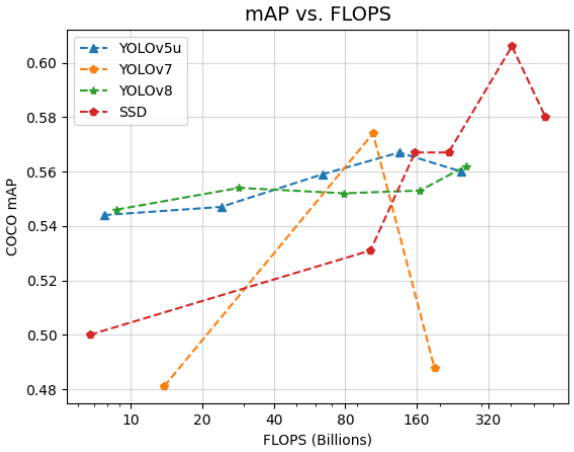

Fig. 1: Graph showing the mAP of the 4 most promising detector families depending on their complexity. For low to middle complexity YOLO detectors are better while for very high complexity SSD is the better choice.

Based on these metrics, the small models *YOLOv5-nu* and *YOLOv8-n* are on par and best suited for edge devices. The best performance overall was demonstrated by *SSD ResNet50 v1 FPN* which is complexity wise on the other end of the spectrum. For detectors in the middle of the complexity spectrum of this experiment, *YOLOv7* reached the highest score in mAP and AP and also a high position in the mAR ranking, making it a model for further investigation.
Comparing the mAP results with these of COCO, all models, with exception of *YOLOv7-x* and *EfficientDet*, are exceeding the mAP score of at least 0.02. The biggest difference can be observed for the *YOLOv5-n* and *YOLOv5-nu* models which exceed their COCO mAP score by about 0.2. One reason is that the used dataset is not as divers as COCO, making it easier to adept from train to test images. The training and testing set are very similar in general, with same backgrounds, recording altitude and illumination. Due to the perspective and the mostly white colour of the sheep, they stand out against the mostly greenish background, without visible pose and size differences, but mostly by rotation. Even though 81.9 % of the test dataset instances are large objects based on the original images, for inference, when down scaled to a edge length of 640 pixel, 90.6 % of the objects are categorized as small and the rest as medium sized. This underlines that the used dataset is not very diverse regarding object sizes. In addition to the above-mentioned problems of the dataset, artefacts such as motion blur, overexposure or noise, which are to be expected in real applications, are almost completely absent.

Looking at the visualized detection results in example for the *YOLOv8-l* model, some patterns can be discovered. One thing is that the detection works quite well when the sheep are seen from above, but gets worse when the sheep are seen more from the side, as to be seen in Figure 2. Another observation is that there are more conspicuous false detections at lower altitudes, as shown in Figure 3. This seems to be less likely in images from higher flight altitudes. For images in which sheep form a flock and have small distances to each other, bounding boxes sometimes include multiple sheep partially without including one whole sheep. In these cases it also gets clear that rotated bounding boxes should be considered over the normal horizontal bounding boxes.

5 Conclusion and Future Work

In this paper we introduced the main ideas and objectives of the European project *SPADE* in which we will create a multi-purpose physical-cyber agri-forest drones ecosystem for governance and

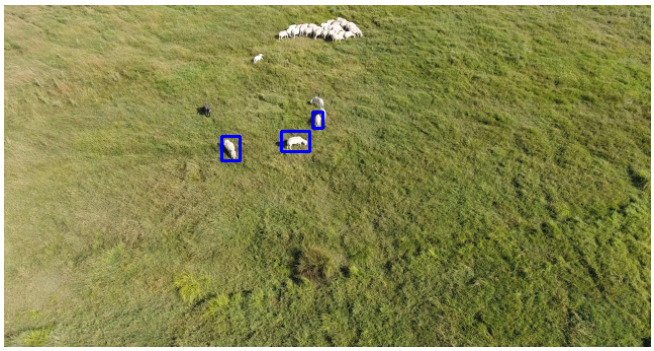

Fig. 2: Visualized results for the YOLOv8-l model. Not-detection of sheep when seen from side and more far away.

Fig. 3: Visualized results for the YOLOv8-l model. Miss detection at low flight altitude. 4 false positives on 19 true positives.

environmental observation. While the project serves three main use-cases, namely agriculture, forestry, and livestock, we focused on the latter by training and evaluating a set of different state-of-the-art object detectors on a publicly available image dataset of sheep filmed from UAVs. Therefor, we were able to get a first impression how well general object detectors perform for the task at hand and get a better understanding of the capabilities as well as shortcomings of each detector.

Although, in this paper the main focus was on the livestock use-case, in the following sections we give an overview of future work in all three use-cases.

5.1 Livestock Management

Keeping in mind that the overall objective of *SPADE* is that as much as possible should be running on the drone directly, in future work we concentrate more on lightweight object detectors which are capable of delivering real-time performance on low-power edge devices. In order to achieve satisfying results in real-world environments, we therefore plan to first train selected object detectors on large datasets of drone images, such as the VisDrone dataset [38] for instance, and then transfer-learn the resulting models on aerial datasets of different sheep species. Moreover, object detection from drone images poses several challenges compared to object detection from ground-level images. For instance, different altitudes of drones result in large scale difference that makes it challenging to accurately detect and localize objects. Additionally, UAVs are subject to vibrations and movements during image capture, which can introduce blur and motion artifacts in the images. These artifacts can degrade the quality of the images, making object detection more challenging. To overcome these issues, we will investigate different augmentation techniques which can be used during training to enhance the robustness of object detectors. In parallel we will investigate techniques that were especially designed for the detection of oriented objects and object detection from aerial images, which we briefly reviewed in Section 2.

Furthermore, real-time object tracking is a vital component of UAV-based applications. Therefore, we plan to apply the resulting object detectors to video instead of still images and additionally combine them with methods for multi-object tracking. Traditional tracking algorithms such as Kalman Filters and Particle Filters as well as more advanced deep learning-based trackers such as DeepSORT [34], StrongSORT [9], ByteTrack [36], and CenterTrack [37], enable UAVs to continuously track and follow objects of interest in real-time, even in complex and dynamic environments. Continuous multi-object tracking and object association in consecutive frames builds the basis for subsequent tasks such as motion analysis for instance. By tracking objects over time, we hope to go beyond simply counting the number of sheep that are visible, but additionally analyze their motion patterns, speed, direction, and trajectories. This information is valuable for tasks such as activity recognition, behavior analysis, and anomaly detection.

Because of the lack of available video datasets and to be able to benchmark different types of object detectors and tracking paradigms, we already recorded a set of videos from drones showing herds of sheep on the University Farm of Aristotle University of Thessaloniki (A.U.Th.). The dataset currently contains 17 videos of different lengths, ranging from 30 seconds to over 4 minutes. We are currently in the process of annotating the dataset using the

Table 2 Detection results

Detector	mAP	AP	mAR	Input	FLOPS (B)
YOLOv5-n	0.475	0.904	0.541	640	7.7
YOLOv5-s	0.503	0.933	0.571	640	24.0
YOLOv5-m	0.538	0.934	0.604	640	49.0
YOLOv5-l	0.55	0.936	0.611	640	109.1
YOLOv5-x	0.57	0.945	0.63	640	205.7
YOLOv5-nu	0.544	0.936	0.61	640	7.7
YOLOv5-su	0.547	0.945	0.616	640	24.0
YOLOv5-mu	0.559	0.945	0.629	640	64.2
YOLOv5-lu	0.567	0.955	0.639	640	135.0
YOLOv5-xu	0.56	0.946	0.632	640	246.4
YOLOv6-n	0.446	0.889	0.51	640	11.4
YOLOv6-s	0.516	0.933	0.582	640	45.3
YOLOv6-m	0.534	0.944	0.607	640	85.8
YOLOv6-l	0.548	0.946	0.615	640	150.7
YOLOv7-tiny	0.481	0.921	0.547	640	13.8
YOLOv7	0.574	0.955	0.637	640	104.7
YOLOv7-x	0.488	0.919	0.561	640	189.9
YOLOv8-n	0.546	0.936	0.612	640	8.7
YOLOv8-s	0.554	0.945	0.625	640	28.6
YOLOv8-m	0.552	0.943	0.623	640	78.9
YOLOv8-l	0.553	0.944	0.627	640	165.2
YOLOv8-x	0.562	0.944	0.634	640	257.8
SSD MN v2 FPN-lite	0.5	0.926	0.589	640	6.7
SSD MN v1 FPN	0.531	0.939	0.616	640	102.2
SSD RN50 v1 FPN	0.567	0.947	0.645	640	157.4
SSD RN101 v1 FPN	0.567	0.946	0.642	640	218.0
SSD RN50 v1 FPN	0.606	0.959	0.676	1024	402.8
SSD RN101 v1 FPN	0.58	0.934	0.652	1024	558.0
EfficientDet-D0	0.224	0.616	0.317	512	4.5
EfficientDet-D1	0.286	0.701	0.383	640	9.97
EfficientDet-D2	0.387	0.832	0.47	768	18.0
EfficientDet-D3	0.463	0.883	0.534	896	42.6
EfficientDet-D4	0.491	0.92	0.556	1024	97.4
EfficientDet-D5	0.315	0.754	0.394	1280	234.4
EfficientDet-D6	0.0	0.0	0.001	1280	488.8
CenterNet MN v2 FPN OD FT	0.249	0.45	0.285	512	3.9
CenterNet MN v2 FPN OD	0.259	0.461	0.294	512	4.1
CenterNet RN50 v1 FPN	0.488	0.854	0.56	512	59.2
CenterNet RN101 v1 FPN	0.481	0.854	0.554	512	98.0
Faster RCNN RN50 v1	0.383	0.808	0.467	640	210.2
Faster RCNN RN101 v1	0.465	0.847	0.529	640	270.9
Faster RCNN RN50 v1	0.49	0.932	0.58	1024	313.6
Faster RCNN RN101 v1	0.505	0.946	0.592	1024	468.7
Faster RCNN Inception RN v2	0.402	0.843	0.469	640	1655.8

Table 3 Detection results for the test set ordered by detector family and FLOPS. The abbreviations RN and MN stand for ResNet and MobileNet respectively. The input value indicates the edge length of the input image size.

open-source Computer Vision Annotation Tool (CVAT) [5], an open-source web-based tool specifically designed for annotating images and videos to create training datasets for computer vision algorithms. This dataset will serve as the basis for further investigation of sheep detection and tracking in videos. We plan to publicly release the dataset once it is fully annotated and hope it will spark interest to develop novel approaches for detection, classification, tracking, and even identification of animal species in UAV footage.

Besides the detection and tracking of animals we plan to also investigate methods for landscape classification and grassing region detection, since with such an technology farmers can leverage the analysis of vegetation and land characteristics to optimize their grazing management practices. This includes strategically allocating grazing areas and maximizing the availability of high-quality forage for livestock. Thus, by utilizing this technology, farmers gain the ability to make informed decisions that improve productivity, enhance animal welfare, and promote environmental sustainability on their farms. We outlined first attempts for that task in Section 2.4 but have the opinion that this needs further research, since most publications in landscape classification focus on multi-spectral HRSI instead of using plain RGB images and such expensive cameras might not be available in every situation.

5.2 Forestry

UAVs in combination with state-of-the-art computer vision algorithms implemented on edge devices can enable real-time monitoring of forest ecosystems, including tree health, growth patterns, and environmental conditions. This information can help forest managers to make informed decisions regarding forest management practices, such as identifying areas prone to disease or pests, optimizing harvesting strategies, and assessing the impact of climate change.

In order to achieve these goals, we aim to combine the most effective and efficient object detectors, introduced above with algorithms for object instance segmentation to analyse trees. In contrast to object detection, which draws rough bounding boxes around each located object and classifies to which class the object belongs to, instance segmentation goes beyond that by accurately outlining the boundaries of each individual object, obtaining a pixel-wise prediction of the object's boundaries.

5.3 Agriculture

Besides livestock management and forestry, precision agriculture can greatly benefit from modern computer vision algorithms implemented on UAVs. This enables farmers to monitor and manage crops more effectively, optimizing resource usage, such as water and fertilizers, and identifying areas requiring attention, such as pest infestations or nutrient deficiencies.

Also, the agriculture use-case would greatly benefit from algorithms for tree detection and segmentation on edge devices. However, additional to that the automatic detection and classification of pests and diseases of trees and plants would be of high interest. Several researchers have already worked on the automatic detection of fruit flies [8, 15]. However, transferring such algorithms to edge devices is not a trivial task and needs further research.

6 Acknowledgments

Funded by HORIZON Europe HE-2022: SPADE – 101060778 ©2023 IEEE. We also wish to extend our appreciation to Professor Bossis of Aristotle University of Thessaloniki (https://www.auth.gr/, http://www.agroctima.auth.gr/en/) and his team for organizing the first SPADE Livestock Trial.

7 References

1 Abdelmalek Bouguettaya, Hafed Zarzour, Ahmed Kechida, and Amine Mohammed Taberkit. Deep learning techniques to classify agricultural crops through uav imagery: A review. *Neural Computing and Applications*, 34:9511–9536, 2022.

2 Babak Chehreh, Alexandra Moutinho, and Carlos Viegas. Latest trends on tree classification and segmentation using uav data —a review of agroforestry applications. *Remote Sensing*, 15(9), 2023. ISSN 2072-4292. doi: 10.3390/rs15092263. URL https://www.mdpi.com/2072-4292/15/9/2263.

3 A. Chen, M. Jacob, G. Shoshani, and M. Charter. Using computer vision, image analysis and uavs for the automatic recognition and counting of common cranes (grus grus). *Journal of Environmental Management*, 328, 2023.

4 Chungling Chen, Ziyue Zheng, Tongyu Xu, Shuang Guo, Shuai Feng, Weixiang Yao, and Yubin Lan. Yolo-based uav technology: A review of the research and its applications. *Drones*, 7, 2023.

5 CVAT.ai Corporation. Computer Vision Annotation Tool (CVAT), September 2022. URL https://github.com/opencv/cvat.

6 Meyer E De Kock, Václav Pohůnek, and Pavla Hejcmanová. Semi-automated detection of ungulates using uav imagery and reflective spectrometry. *Journal of Environmental Management*, 320:115807, 2022.

7 Fabricio de Lima Weber, Vanessa Aparecida de Moraes Weber, Pedro Henrique de Moraes, Edson Takashi Matsubara, Debora Maria Barroso Paiva, Marina de Nadai Bonin Gomes, Luiz Orciria Fialho de Oliveira, Sergio Raposo de Medeiros, and Maria Istela Cagnin. Counting cattle in uav images using convolutional neural network. *Remote Sensing Applications: Society and Environment*, 29:2352–9385, 2023.

8 Yoshua Diller, Aviv Shamsian, Ben Shaked, and Yam Altman. A real-time remote surveillance system for fruit flies of economic importance: sensitivity and image analysis. *Journal of Pest Science*, 96:611–622, 2023.

9 Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. Strongsort: Make deepsort great again. *IEEE Transactions on Multimedia*, 2023.

10 Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. pages 6569–6578. IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

11 Chong Fan and Ru Lu. Uav image crop classification based on deep learning with spatial and spectral features. *IOP Conference Series: Earth and Environmental Science*, 783, 2021.

12 Ross Girschick. Fast-rcnn. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015.

13 Ross Girshick. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

14 J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3296–3297, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society. doi: 10.1109/CVPR.2017.351. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.351.

15 Renjie Huang, Tingshan Yao, Cheng Zhan, Geng Zhang, and Yongqiang Zheng. A motor-driven and computer vision-based intelligent e-trap for monitoring citrus flies. *Agriculture*, 11(5):460, 2021.

16 Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, Zeng Yifu, Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, November 2022. URL https://doi.org/10.5281/zenodo.7347926.

17 Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, January 2023. URL https://github.com/ultralytics/ultralytics.

18 Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

19 Chuyi Li, Lulu Li, Yifei Geng, Hongliang Jiang, Meng Cheng, Bo Zhang, Zaidan Ke, Xiaoming Xu, and Xiangxiang Chu. Yolov6 v3.0: A full-scale reloading, 2023.

20 Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

21 Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, and Cheng-Yang Fu. Ssd: Single shot multibox detector. *CoRR, abs/1512.02325*, 2015.

22 Godfrey Nolan. Sheepcounter dataset, 2021. URL https://universe.roboflow.com/riisprivate/sheepcounter.

23 Stefano Puliti and Rasmus Astrup. Automatic detection of snow breakage at single tree level using yolov5 applied to uav imagery. *International Journal of Applied Earth Observation and Geoinformation*, 112: 102946, 2022. ISSN 1569-8432. doi: https://doi.org/10.1016/j.jag.2022.102946. URL https://www.sciencedirect.com/science/article/pii/S1569843222001431.

24 Stefano Puliti, J Paul McLean, Nicolas Cattaneo, Carolin Fischer, and Rasmus Astrup. Tree height-growth trajectory estimation using uni-temporal UAV laser scanning data and deep learning. *Forestry: An International Journal of Forest Research*, 96(1):37–48, 07 2022. ISSN 0015-752X. doi: 10.1093/forestry/cpac026. URL https://doi.org/10.1093/forestry/cpac026.

25 Kristina Rančić, Boško Blagojević, Atila Bezdan, Bojana Ivošević, Bojan Tubić, Milica Vranešević, Branislav Pejak, Vladimir Crnojević, and Oskar Marko. Animal detection and counting from uav images using convolutional neural networks. *Drones*, 7(3):179, 2023.

26 Joseph Redmon. You only look once: Unified, real-time object detection. *CoRR abs/1506.02640*, 2015.

27 Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time oject detection with region proposal networks. *CoRR, abs/1506.01497*, 2015.

28 B Shaked, A Amore, C Ioannou, F Valdés, B Alorda, S Papanastasiou, E Goldshtein, C Shenderey, M Leza, C Pontikakos, et al. Electronic traps for detection and population monitoring of adult fruit flies (diptera: Tephritidae). *Journal of Applied Entomology*, 142(1-2):43–51, 2018.

29 Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. pages 6105–6114. International Conference on Machine Learning (ICML), 2019.

30 Mingxing Tan, Ruoming Pang, and Quoc v. Le. Efficientdet: Scalable and efficient object detection. Computer Vision and Pattern Recognition (CVPR), 2020.

31 Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

32 Jing Wang. Landscape classification method using improved u-net model in remote sensing image ecological environment monitoring system. *Advanced Big Data Analysis Technologies for Environmental Monitoring Data*, Article ID 9974914, 2022.

33 Y. Wang, L. Ma, Q. Wang, N. Wang, and D. Wang. A lightweight and high accuracy deep learning method for frassland grazing livestock detection using uav imagery. *Remote Sensing*, 15, 2023.

34 Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. IEEE International Conference on Image Processing (ICIP), 2017.

35 Xin Wu, Wei Li, Danfeng Hong, Ran Tao, and Qian Du. Deep learning for unmanned aerial vehicle-based object detection and tracking: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 10:91–124, 2022.

36 Yifu Zhang, Peize Sun, Yi Joang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Wenyu Luo, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. European Conference on Computer Vision (ECCV), 2022.

37 Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. European Conference on Computer Vision (ECCV), 2020.

38 Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. doi: 10.1109/TPAMI.2021.3119563.

39 Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao. Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios. IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021.